PATENT APPLICATION

CLUSTERING USER INTERFACE

Inventor:

> Peter Sklar
> a citizen of U.S.A.
> 837 Cheltenham
> Santa Barbara, CA   93105

Assignee:

> NONE

Entity: SMALL

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California   94111-3834
(415) 576-0200

CLUSTERING USER INTERFACE

5

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of U.S.
Patent Application No. 08/711,300, filed September 6, 1996. *now U.S. Patent No. 5,790,121*
That patent application is incorporated by reference herein
10   for all purposes.

*RB
23 Oct
1998*

BACKGROUND OF THE INVENTION

This invention relates to computer database
searches. More precisely, this invention provides an improved
15   interface for performing computer database searches and
filtering search results.

In the typical database search, a user queries the
database by selecting a set of criteria (a request) and
submitting those criteria to a database engine. The criteria
20   might be in the form of a range of allowable values for a
given field, an upper limit, a lower limit, or an exact match.
Multiple field criteria can be combined by the use of logical
operators (e.g., NOT, AND, OR, GREATER_THAN). The criteria
might also include comparisons between multiple fields (e.g.,
25   AGE >= IQ). Once the criteria are submitted to the database
engine, the database engine selects all the records in the
database that meet all of the criteria selected by the user
and returns those records to the user.

Many different methods have been utilized to
30   facilitate the user's creation of database requests. One
standard for specifying database search criteria to the
database engine is Structured Query Language (SQL). SQL
statements are strings of text and numbers which define the
search request. If the end user requesting data from a
35   database is proficient in SQL, the end user can specify the
SQL statement directly. However, where the end user is not
proficient in SQL programming, a user interface might be
provided to allow the end user to intuitively select elements

from the user interface, which are then converted into SQL statements for submission to an SQL-capable database engine. Many user interfaces which convert user input into SQL statements are known. In some interfaces, a user answers a
5   series of questions, fills out an on-screen form or selects from a finite number of choices using a mouse or other pointing device.

Once such search request is submitted, the database engine returns the records meeting the criteria, if any, and
10   the user interface displays the records or some indication of the records. Several methods have been used to display the data returned from the database query, such as tables, charts, graphs, and graphic images such as maps or tree structures. The term "maps" as used herein can be geographical maps or
15   logical maps laying out data points in an N-dimensional coordinate system.

It is also known to allow for refinement of a database search after the results of an initial search have been obtained. A refinement search is often desired by users
20   where the initial search produces too many or too few records. With a refinement search, users might edit the initial search request or add additional criteria to the initial search request. By doing so, users can find the proper quantity and quality of records they need.

25   An important factor in determining a proper search criteria is prior knowledge of the database and the distribution of the values in each field. For example, the user knows that the database contains only a listing of Democrats, a search for male Republicans over the age of fifty
30   living in cities of at least one million people will turn up no records. The user might then waste time refining the search by first eliminating any limitation on the size of the city, then eliminating the age limitation, eventually to discover that there are no Republicans in the database. The
35   opposite result, too many records, could occur if the user specified too general a set of search criteria. In that case, the user might also waste time iteratively narrowing the

search with little effect. Often, the success of a database search is dependent upon luck.

Another shortcoming of most current search methodologies is that users do not gain any knowledge during the search process to help refine their search. In some systems, database records, especially geographical databases, present records to the user as a set of dots overlaid on a map. While such an interface might be useful for a few dots, it becomes impractical for use with a large number of dots.

With large numbers of dots, either the dots will be too small to be selected with a pointing device or, if large enough, the dots will obscure other dots. FIG. 1 illustrates the former problem; FIG. 2 illustrates the latter.

One way to avoid the selection problem is to allow the user to click the pointing device near the point of interest and treat the click as the selection of all points within a radius ("a radius search"). Of course, this has the disadvantage of selecting too many records, too few records or the wrong records.

Another way to avoid the selection problem is to have a user select on arbitrary boundaries such as state boundaries, and gets a text listing of all the locations in that state, and possibly a further graphical selection of a county or region. The state selection is shown in FIG. 3. The disadvantage of this approach is that the user, at most, knows only whether or not a state contains at least one dot.

In addition to geographical data and other data which can be plotted in a two-dimensional plane (or an N-dimensional coordinate system with N being an integer greater than zero), data in searchable hierarchical structures often need to be searched.

One way to present the data points in a hierarchical structure search is to present all the data points. However, with large numbers of data points, this is impractical.

For example, Microsoft Word™ word processing

software provides a feature for searching for files among the hierarchical directory structure of a disk. The matching files are displayed in a directory tree structure showing each of the directors for the files along with their parent

5   directories. While this is useful for a small number of files, the display for a large number of files would not fit on the screen and thus the user must scroll through the listing or manually "collapse" uninteresting directory structures to be able to see the directory structures of

10 interest. This approach does not give the user a "big picture" view of file structures, unless the user knows what the big picture looks like and creates it for themselves.

Yet another shortcoming of prior art information displays appears when there are more items to be listed than

15 can fit on a display. One solution is to list all the items on a scrollable list. The other is to group the items into categories and display categories first, allow the user to select one of the categories, then display the items matching the selected category. Neither of these approaches is

20 entirely satisfactory, however. In the former case, too much data and not enough information is presented. In the latter case, depending on the categories, the display might be underused.

Therefore, what is needed is an improved search

25 interface which presents the user with information and views of the overall data being searched, in order to allow for an informed search.

SUMMARY OF THE INVENTION

30 In one embodiment of a search interface according to the present invention, a user is presented with a display map from which the user selects database records of interest. For the records which are in range of the display map's limits, a cluster evaluator groups some of the records into clusters.

35 Generally, a cluster is a set of records which would be clustered on the display map. On the display map, unclustered records are represented by item icons, while clusters are represented by cluster icons. If a user selects an item icon,

the associated record is selected and a predetermined action is taken. If a cluster icon is selected, the display map is "zoomed in" to show greater detail around the cluster. Typically, a number of clustered records become unclustered

5 records when the zooming occurs, because the clustering criteria are sharpened. The user can iteratively select clusters, resulting in greater and greater detail, until the user selects an item icon and the predetermined action is taken with that record. The predetermined action could be any

10 action which might be taken with an individual record, such as displaying additional fields for the record, initiating a process for manipulating the record, or the like.

The term "icon," as used herein, refers to a user interface element which is presented to the user and is

15 selectable by the user using a pointing device. Depending on context, icons can be regular shapes or irregular shapes, can have identified borders or implied borders, and can either be illustrated with a graphic that hints to the meaning of the icon or can be a plain area whose meaning is identified by

20 context.

The records are mapped to the display based on field values of the record. For example, a two-dimensional map could be used to display records which contain longitude and latitude values. The same principles could be used for

25 one-dimensional data, three-dimensional data, or higher dimensional data which can be suitably represented on a two- or three-dimensional display. If the search data is numerical data from as many fields as the display has dimensions, cluster icons are graphically displayed as shaded regions on a

30 graph or map, roughly coextensive with the mappings of the records to the display. Unclustered records are graphically represented by item icons at the location corresponding to the field data of the records. Thus, a user can easily see where the clusters are, where the unclustered records are, and

35 regions where no records exist. Clusters might overlap, so that one record is clustered into more than one cluster. By selecting a particular cluster, the user reduces the dataset to a subset which contains only those records which were

clustered into the selected cluster. The new subset is then clustered to allow the users to narrow their search further, and the selection process to begin again. The users can end their search at any time, but eventually, the dataset will be

5   reduced to a set of records that do not cluster.

The present invention can also be used to present hierarchical structures for user selection of records from such structures. Examples of such hierarchical structures include electronic file storage directory structures,

10   organizational charts within large organizations, plain hypertext pages on a Web site, or an intranet. In such cases, clustering is done by using a metric which is the number of links rather than a distance in a coordinate system or a weighted metric which takes into account hierarchies where

15   some links are more important than others, such as taxonomy hierarchy. For example, two files in the same directory would be deemed to have a metric between them of 0 and a file in a parent directory would have a metric between a file in an immediate subdirectory of 1. For files which are "cousins,"

20   the metric could either be the number of links in the shortest path between the two files (2) or the number of links to a common ancestor (1). Such a clustering process might prove to be highly useful in sorting through corporate data, if it is clustered and organized according to an organizational chart,

25   wherein information provided by departments nearer to the user's department is given more prominence than information from departments which are "farther" away on the organization chart. Presentation of hierarchical information would also be greatly simplified using the present invention, as many search

30   engines or searching Worldwide Web sites tend to either show too many hits or too few hits. For example, a large URL (uniform resource locator) database such as YAHOO!'s database (http://www.yahoo.com/) would provide too many references matching the word "team" to be of use to a searcher. However,

35   with the clustering user interface of the present invention and a starting point, such as the "Oakland Raiders" subcategory of Yahoo!'s hierarchical subject database, the user could be presented with all the hits, where many of the

hits are clustered into a single entry. The user would be provided with a cluster icon to "open" that cluster, possibly leading to other clusters. Instead of a URL database, the invention could also be applied to a "yellow pages" database.

5   In order to achieve a manageable number of clusters the clustering algorithm can incrementally increase the minimum distance separating two clusters. As the distance increases, the number of clusters will decrease. When the total number of clusters becomes less than a predetermined

10   threshold, that distance is used in the clustering algorithm. An alternative method for incremental clustering would be to initially create clusters of records that are separated by very small distances and then gradually combine these clusters into bigger clusters until the total number of clusters

15   becomes less than the threshold. In order to prevent creating too few clusters, the clustering process can be modified to terminate before all cluster pair distances are calculated. For instance, if a current step of clustering is combining all clusters less than five miles apart, the clustering process

20   could terminate when the goal of getting the total number of clusters of less than twenty has been achieved even though there are still clusters less than five miles apart remaining.

       In yet another application of clustering, a list of categorized items are displayed as a list of labels, where a

25   label is a textual and/or graphical indication of what the label represents. Each label represents either an item or a category, where an item label represents an individual item, while a category label represents an entire category of items. A display manager determines which items are shown

30   individually and which are grouped by category so that the full display is used. One method of filling a display is to set a threshold count and to tag for individual display all items in categories having no more than the threshold count of items and to tag for category display the items in categories

35   having more than the threshold count of items. The threshold count is then adjusted so that the labels just fill the display. Thus, if there are too many labels, the threshold count is raised, and if there are too few labels, the

threshold count is lowered, until the threshold count reaches one, at which point the display will show all of the items with item labels.

A further understanding of the nature and advantages of the inventions herein may be realized by reference to the remaining portions of the specification and the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a clickable map with small squares on the map representing individual records in a database of records where each record has a location within the map of the continental United States.

~~FIG. 2 is a clickable map where a state of interest is selected.~~

FIG. 3 is a clickable map ~~as used in the prior art~~ where a state of interest is selected.

FIG. 4 is a front view of a computer system which could be used to implement the present invention.

FIG. 5 is a block diagram of the components of the computer system shown in FIG. 4 as they relate to the present invention.

FIG. 6 is a clickable map according to the present invention with unclustered item icons and cluster icons are overlaid on a map.

FIG. 7 is a clickable map which results from user selection in FIG. 6 of the cluster under the mouse cursor.

FIG. 8 is a flowchart of a display process according to the present invention.

FIG. 9 is a block diagram of a networked system over which the present invention could be used.

FIG. 10 is a map of a directory structure which can be searched using the present invention.

FIG. 11 is a block diagram of a display in which items are represented by item labels and category labels.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention can operate on a wide variety of computers with a number of suitable configurations, each being the preferred embodiment for a particular implementation. For example, a palmtop implementation might be preferred for use with a restaurant finding device designed to be used by a person walking around looking for a good restaurant meeting the person's criteria. A desktop model, with high resolution graphics and a direct Internet connection might be preferred for an office-based application. This detailed description of the invention will describe an embodiment using a desktop computer and an embodiment using the Internet (the global internetwork of networks generally referred to by that name), although it should be understood that many other configurations of computing devices might also be used to provide similar functionality. For example, the user might be an Internet or Intranet surfer, a system administrator, an email administrator or an interactive television viewer/user. The computing device might also be a mainframe terminal or computer kiosk.

FIG. 4 shows a typical desktop computer 10 made up of a system box 12 enclosing a hard disk 14, a keyboard 16, a mouse 18, and a monitor 20 including a display surface 22. As is well known in the art, desktop computer 10 can be programmed with programs stored on hard disk 14 to perform various tasks. These tasks might involve accepting user input from keyboard 16 and mouse 18, and/or displaying images or text on display surface 22. The input from mouse 18 is in the form of either movement in two dimensions (rolling the mouse) or clicking (pressing one of the mouse buttons). It is also well known to replace mouse 18 with equivalent two-dimensional pointing devices, or even three-dimensional pointing devices. With suitable software or hardware, monitor 20 can be made to display three-dimensional images, as is well known in the graphic arts.

FIG. 5 is a block diagram illustrating the typical contents of system box 12 in more detail. A central processing unit (CPU) 202 executes programs stored in a

program memory 204 by reading program instructions from program memory 204 via processor bus 206. Of course, multiple CPU's could be provided for faster operation in some computer systems. CPU 202 interprets input signals from mouse 18

5   provided on bus 206 after being suitably transformed by a mouse driver 208 which might be hardware, software, or a combination of both. CPU 202 uses a RAM (random access memory) 210 for storage of program variables as needed. Bus 206 also provides a path for CPU 202 to send output signals to

10   display surface 22 of monitor 20 (see FIG. 4) via a display driver 222 and video display memory 224. If used, desktop computer 10 can communicate over a network with a network connection 218.

       To program desktop computer 10 according to the

15   present invention, hard disk 14 is loaded with a map database 220 and one or more programs. To begin executing a program which implements the present invention, CPU 202 causes the program to be loaded from hard disk 14 into program memory 204. In a typical personal desktop computer, program memory

20   204 and RAM 210 are different segments of one memory structure. As shown in FIG. 5, the program includes a cluster evaluator 230, a display generator 232, an input processor 234, a database query interface 236 and a record action module 238. Other components, such as a database engine, are

25   included but not shown in FIG. 5.

       Using the system described in FIGS. 4-5, a user will select records from map database 220 and have a selected record acted on by record action module 238. A process of using the system according to the present invention is

30   described in more detail in connection with FIGS. 6-7. First, two examples of the prior art will be described. In both the prior art and the presently described inventive embodiment are user interfaces in which the user's goal is to select a record according to geographical location. One use of such a system

35   is to allow users to select one store of a national chain of stores and perform an action with respect to that store (e.g., dial that store's telephone number, get a map of surface

streets giving directions to the store, get listings of hours of operation, products carried, manager name, etc.).

FIG. 1 shows one method of record selection known in the prior art. Each of the small black squares on the map

5   represents one store. To select a store, the user positions a mouse cursor 250 over the desired one of the black squares and clicks a mouse button to select the store. While that may be useful in areas where there are few stores, such as Colorado and Oklahoma, it would be very difficult for the user to

10   position the mouse exactly to select the desired store if it were in California, Washington, Texas or the East Coast. The selection difficulty increases if the display uses, instead of small black squares, icons displaying the company's logo.

FIG. 2 shows what a user interface might look like

15   if icons displaying a company's logo were to be presented in each dot. As Fig. 2 shows, the logo icons would be completely unwieldy even with only a few locations.

FIG. 3 shows one way to avoid the problems illustrated in FIGS. 1-2. Instead of the user having to

20   select one store from all the national stores, the user instead selects the state of interest. In FIG. 3, the user has positioned the cursor and selected the state of Wyoming. In response, the computer would show an expanded, zoom display of the state of Wyoming. As could have been seen from FIG. 1

25   or FIG. 2, the selection of Wyoming is fruitless because there are no stores in Wyoming. On the other hand, the user might have selected California, which does have many stores. However, the computer might need to zoom in one more level if too many stores are represented in the state view.

30   FIG. 6 shows an improved display 300 according to the present invention. Overlaid on a map of the contiguous United States, display 300 includes item icons 302 and cluster icons 304. Item icons 302 represent the locations of individual stores which are not clustered with other stores as

35   determined by cluster evaluator 230. Using mouse cursor 310, the user can select either a cluster icon 304 or an item icon 302. If the user selects an item icon 302, the computer responds with the predetermined item action. If the user

selects a cluster icon, such as the one shown under mouse
cursor 310, the computer zooms in to the map shown in FIG. 7.
As should be apparent from FIG. 6, the user can easily
identify where there are no stores and where there are

5   collections of stores. Note that the data and the cluster
icons are not necessarily constrained by state or other
boundaries. The shape of the cluster icons are determined by
the locations of their constituent items, as explained below.

     Referring now to FIG. 7, the zoomed in map 320 now

10   shows several individual stores in a second layer of clusters.
The process of zooming continues until the user aborts or
selects an item icon. Note that, in FIGS. 6-7, the individual
icons can be large enough to show the company's logo ("K" for
the imaginary sunglass retailer "Kool Shades" used for this

15   example) without interfering with the selection process.

     The display maps shown in FIGS. 6-7 are generated by
computer system 10 according to the process shown in the
flowchart of FIG. 8. That flowchart includes a series of
steps numbered in ascending order which are executed in that

20   order except where indicated. In the first step (S1), the
computer determines the map bounds. For a national chain such
as Kool Shades, the default initial map bounds are defined by
the rectangle enclosing the contiguous United States. In step
S2, the computer determines which records are within the map

25   bounds. These are the records the user can select among.

     Of the selectable records, cluster membership is
determined (S3). Each cluster has a set of records which are
its members. In the illustrations provided here, each record
is either a member of no cluster or a member of one cluster.

30   However, the system could be designed to allow a record to be
in more than one cluster, so long as each cluster contains a
subset of records significantly smaller in size than the set
of all records. Many of the different clustering algorithms
in different fields of math and science will function

35   adequately for this purpose, depending on the goals of the
search and type of data. Other clustering factors, such as
the metric used and the normalization of metrics, play
important roles in the search process.

One method for clustering uses a geographic distance metric. Each record for a Kool Shades store includes the longitude and latitude of the store. Using this information, the geographic distance between two stores is easily

5 calculated using known techniques.

With other types of data, a metric can be generated from the values for the data. For example, if the user were selecting records not based on a geographic location, but based on the records' "locations" on a plot of two field

10 values, those field values might need to be normalized. One way to normalize the values is to scale each axis of the plot so that the values fall within a square. This is equivalent, for a first field and a second field, to dividing each first field value by the range of first field values and dividing

15 each second field value by the range of second field values.

Although not required for distance calculations, the values can be transformed so that all normalized values fall between zero and one. Thus, a new value, V', for a field value can be calculated from the original field value, V, with

20 the formula:

$$V' = K(V-L)/(H-L)$$

where L is the lowest field value and H is the highest field

25 value. The multiplicative constant K is set to one to keep all values in the range from zero to one, but it could be set to any other suitable value.

In any case, not only can a position in the two-dimensional display space can be found for any record, the

30 "distance" between any two records can be found. These distances are used to determine which records cluster with which other records. One method for clustering data is to consider each record as an N-dimensional sphere (a circle in the case of Kool Shades stores). Any records whose spheres

35 intersect would be clustered together. The radius of the spheres may depend on the needs of the application, the total number of records, the distance between map boundaries and the density of records in particular locations.

In the case of exclusive clustering (each record is a member of at most one cluster), the following pseudocode describes the process of assigning records to clusters:

```
Main Program: Find All Clusters
        Begin Loop (for each Record)
                If Record is Unclustered then Begin
                        Create New Cluster
                        Add_Record(Record, New_Cluster)
10                      End If Then
        End Loop
End Program


Procedure Add_Record(Record, Cluster)
15          Put Record in Cluster
            Begin Loop (for each Unclustered Record)
                    If(Distance(Record, Unclustered Record) < Threshold)
                            then Add_Record(Unclustered Record, Cluster)
            End Loop
20  End Procedure
```

The result of this process is that each unclustered record will be assigned to one cluster and any record which is within a sphere radius of a clustered record will be a member of that cluster. As should be apparent, the size of the cluster is often dependent on the distribution of its members. In the above program implementation, all records are assigned to clusters, with the unclustered records each being the sole member of their cluster. Although, technically, they may be assigned to such solo clusters, they are considered to be unclustered.

Once the clusters and unclustered records are determined, the cluster icon shapes are calculated (S4). Of course, if the data is not graphically displayable, this step need not be done. A cluster icon can just be a shaded area covering the union of the circles for each record in the cluster. An algorithmically easier method for determining shape is to consider each record as a square or a cube. The

shape of the cluster is then derived by using polygons to outline the shape of the union of the squares or cubes.

In step S5, the map is overlaid with cluster icons and item icons. In step S6, the map and overlays are displayed to the user. As explained above, the user can easily see where unclustered records are, where clusters are, the clusters' extents and the areas where no records are to be found.

In step S7, user input is accepted and in step S8, the type of user input is evaluated. If the user clicks on the map outside an item icon or a cluster icon, the program flow is from step S8 back to S7 (i.e., no action is performed). If the user selects an item icon, the predetermined action for items is performed (S9), be it displaying a street map with directions to the individual store or other data about that individual record. If the user selects a cluster icon, the program proceeds to step S10, where a new, zoomed map is generated. Once the new map is generated, the program loops back to step S2 to generate a new set of included records and the program repeats steps S2 through S8. Where applicable, the predetermined action could be a display of the record in detail showing all field values, presenting an editing screen for updating the record, initiating a phone call or an e-mail message to a place or person referred to in the record, or anything else that could be done with just one record.

A concrete example is presented here for further explanation of the process. Suppose data fields to be clustered were latitude and longitude and the cluster icons and item icons were to be displayed on a 400 x 400 pixel map of the contiguous United States. A radius of 8 pixels could then be used to cluster 1000 records scattered over a map of the United States. Since the scale on the US map is roughly 10 miles/pixel, records within 80 miles of each other would be clustered (i.e., the initial cluster threshold would be 80 miles). If the user selects a cluster and the zoomed in map has a resolution of roughly 2 miles/pixel, records within 16 miles of each other will then be clustered. The result of the

larger scale is that more locations are separated and more sub-clusters are created. Of course, with dynamic thresholding, the threshold distance is adjusted to obtain a suitable number of clusters.

5   The basic embodiment of the invention having been described, two variations will now be described. FIG. 9 is a block diagram of an embodiment of a mapping system 800 using an Internet connection. In mapping system 800, a Web browser 802 is used by the user to interact with a Web server 804 via

10 a TCP/IP interface 806 and the Internet 808. Web server 804 provides maps overlaid with icons to Web browser 802, which displays the maps and allows the user to click on points on that map. Web browser 802 returns to Web server 804 and mouse clicks which Web server 804 would need to have to generate a

15 new map. Web server 804 obtains the overlaid maps from a map overlay system 810 which operates substantially as described above, using a map database 812 to obtain maps and store database 814 to obtain records about the location of stores. Of course, the records mapped need not be store records.

20   Initially, map overlay system 810 will request the initial map from map database 812. Typically, map database 812 is a data structure containing map data, but also includes an engine interface whereby map overlay system 810 requests maps by specifying a scale and the four edges of the desired

25 map. As the user selects points on the map, map overlay system 810 will generate new maps according to the process described in connection with FIG. 8.

    Mapping system 800 is useful where the user does not control the map data or the store data. It also allows for

30 easy updating of store locations and other data in store database 814 because it is centralized. Mapping system 800 could also be attached to a company's Web site is such as way that the store location function appears to be part of the company's Web site, but is actually served by a map service

35 provider. This allows for easy updates of map serving software and the underlying maps, since all of that is in one place.

In situations where the clusters and individual records cannot be displayed in a graph or map, the representation of the clusters can be performed by providing the user with summary statistics and summary graphs of the

5 fields and/or combination of fields in each cluster. The user would use this summary information to select his/her desired cluster or record. One such example is shown in FIG. 10, where a file directory structure to be navigated is laid out. Analogous to the location maps, distances between files can be

10 determined, so they can be clustered. The distance between two files might be the number of links to be traversed from one file to the other, or the number of links from a common ancestor of both files to the file furthest from the common ancestor.

15 FIG. 10 shows a selection display for a hierarchical directory structure. The labels shown with cross-hatching are cluster icons for the hierarchical directory list and item icons are shown with gray labels. For example, the cross-hatched label "repair" refers to a directory of that name

20 which contains files meting the search criteria. The file "WINPRINT.DLL" is an individual file meting the search criteria. If the user selects one of the cluster items, a new and expanded display is presented showing the files and clusters underneath the selected subdirectory in more detail.

25 FIG. 10 shows twelve icons, which are either cluster icons or item icons. As each display is generated, the substitution of a cluster icon for a plurality of individual item icons is performed until sufficiently few individual icons remain. Of course, in some cases, such as a single directory having

30 hundreds of files meting the search criteria cannot be effectively clustered using just a file's subdirectory as the cluster parameter.

FIG. 11 illustrates the use of clustering to optimally fill a display 900. Display 900 is a display which

35 might be used to gain insight into a database of cars available for sale. In this particular example, if all the cars for sale were shown, display 900 would list 4911 items, which cannot be shown all at once in a visible, understandable

manner. As should also be apparent, since the cars are grouped into only four categories, limiting the display to only categories, only half of the available display space would be used. With the approach of the present invention, however, some of the items are shown by item labels and others are represented by category labels. In FIG. 11, three item labels 902 and four category labels 904 are shown (however, one of the category labels, 904a, does not represent any items, as all of its items are represented by item labels 902).

The labels 902, 904 which represent items or categories are underlined in the display to indicate to the user that the label can be selected to obtain information about the data which the label represents. Thus, selecting (via a "mouse click" on the item label or other methods) an item label 904 will result in a display of information about the selected item, whereas selecting a category label will result in a display of item labels for items in that category, a request for more information, a listing of items using subcategory labels, or some combination of these.

As shown in FIG. 11, the "Electric" category is opened up to show all of its items, but none of the other categories are opened. If they were, the listing would not fit into display 900, because at least 31 more lines would be used (if the next smallest category, "Antique", is opened). To automatically determine which categories to open, a processor, such as the CPU shown in FIG. 5 and described above, determines how much space is available after the display of the category labels, and then opens categories beginning with the smallest category, until there is no space left in the display to open another category. Alternatively, the processor might set a threshold count and tentatively open each category having a number of items equal to or less than the threshold count and then adjust the threshold count to shrink or expand the list to best fit the display area.

FIG. 11 also shows dynamic data fields 906. These fields are used to display information about an item or a category when a cursor 908 is moved over the item or category.

If cursor 908 is over a category, data fields 906 present summary information about the items in that category, such as the average value for a field in the records for those items. If cursor 908 is over an item, data fields 906 present

5   detailed information about the item. The detailed information can either be the specific values for the same field as used when cursor 908 is over a category or other information.

Using the category labels and item labels, a display area can be efficiently populated, while giving the user an

10   indication of the overall organization of the data in the database.

The above description is illustrative and not restrictive. Many variations of the invention will become apparent to those of skill in the art upon review of this

15   disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the appended claims along with their full scope of equivalents.